

## WHAT IS CLAIMED IS:

1. A method for providing an interface for a function, comprising:  
receiving at least one parameter; and  
in response to receiving the at least one parameter,  
performing an operation relating to a binding on at least one property from at least one of data sources, data source classes, and data-specific implementations of collections and views.
2. The method of Claim 1, wherein performing the operation, further comprises performing handling collection changed events in data collection underlying a collection view.
3. The method of Claim 1, wherein performing the operation, further comprises performing a function relating to array list data collection.
4. The method of Claim 3, wherein performing the function relating to the array list data collection, further comprises performing at least one of the functions within the set comprising: adding; clearing, creating a view, inserting, removing, reversing; setting a range; and sorting.
5. The method of Claim 1, wherein performing the operation, further comprises managing bindings between a dynamic property user interface and source data.
6. The method of Claim 1, wherein performing the operation, further comprises getting the bind declaration object of a binding expression.
7. The method of Claim 1, wherein performing the operation, further comprises performing a collection view function.
8. The method of Claim 1, wherein performing the operation, further comprises implementing a collection view that includes checks for context infinity.

9. The method of Claim 1, wherein performing the operation, further comprises supporting object references to objects being used as data context for a binding.

10. The method of Claim 1, wherein performing the operation, further comprises encapsulating arguments for data transfer events, wherein the events are routed events that are handled by a designated handler based on a delegate.

11. The method of Claim 1, wherein performing the operation, further comprises handling a data transfer event raised by a binding.

12. The method of Claim 1, wherein performing the operation, further comprises representing an object reference to an element, with the object reference being specified by its element ID.

13. The method of Claim 12, wherein the object reference is an explicit object reference.

14. The method of Claim 1, wherein performing the operation, further comprises implementing a collection view for collections based on a list.

15. The method of Claim 1, wherein performing the operation, further comprises serving as a data source for data binding.

16. The method of Claim 1, wherein performing the operation, further comprises holding a collection of named parameters.

17. The method of Claim 1, wherein performing the operation, further comprises representing a single select statement to be submitted to a database.

18. The method of Claim 1, wherein performing the operation, further comprises encapsulating arguments passed in an event relating to at least one of an `ObjectDataSource`, and a `RefreshCompleted` event of an `XmlDataSource`.

19. The method of Claim 1, wherein performing the operation, further comprises handling events relating to at least one of a `ObjectDataSource.RefreshCompleted` event and a `XmlDataSource.RefreshCompleted` event.

20. The method of Claim 1, wherein performing the operation, further comprises receiving a list of sql commands and names of tables that they should be used to fill.

21. The method of Claim 1, wherein performing the operation, further comprises getting data from a SQL Server for use in databinding.

22. The method of Claim 1, wherein performing the operation, further comprises allowing resource reference to a transformer class that is defined as code-behind in a current application.

23. The method of Claim 1, wherein performing the operation, further comprises declaring namespaces to be used in Xml data binding XPath queries.

24. The method of Claim 1, wherein performing the operation, further comprises serving as a data source for data binding to Extensible Markup Language (XML) content nodes.

25. The method of Claim 1, wherein performing the operation, further comprises declaring an individual namespace within an Extensible Markup Language (XML) data source.

26. The method of Claim 1, wherein performing the operation, further comprises managing a view of a data collection.

27. The method of Claim 1, wherein performing the operation, further comprises handing a `CurrentChanged` event raised by collection views, or any class implementing the `ICurrentItem` interface.

28. The method of Claim 1, wherein performing the operation, further comprises representing a method that handles a CurrentChanging event raised by collection view classes, or any class implementing the ICurrentItem interface.

29. The method of Claim 1, wherein performing the operation, further comprises enabling notifications from at least one of the following: items within a collection have changed: an item has been added, removed, or the entire collection has been refreshed.

30. The method of Claim 1, wherein performing the operation, further comprises creating collection view factory classes, which in turn create new CollectionView derived objects.

31. The method of Claim 1, wherein performing the operation, further comprises, maintaining a concept of the current record pointer in a collection view.

32. The method of Claim 1, wherein performing the operation, further comprises creating a One-Way Binding to a Dynamically Updated Data Source.

33. The method of Claim 1, wherein performing the operation, further comprises describing a sort qualifier that is used to sort items in a collection when creating a view.

34. At least one computer-readable medium for providing an interface for a function, comprising:

receiving at least one parameter; and

in response to receiving the at least one parameter,

performing an operation relating to a binding on at least one property from at least one of data sources, data source classes, and data-specific implementations of collections and views.

35. The computer-readable medium of Claim 34, wherein performing the operation, further comprises performing handling collection changed events in data collection underlying a collection view.

36. The computer-readable medium of Claim 34, wherein performing the operation, further comprises performing a function relating to array list data collection.

37. The computer-readable medium of Claim 36, wherein performing the function relating to the array list data collection, further comprises performing at least one of the functions within the set comprising: adding; clearing, creating a view, inserting, removing, reversing; setting a range; and sorting.

38. The computer-readable medium of Claim 34, wherein performing the operation, further comprises managing bindings between a dynamic property user interface and source data.

39. The computer-readable medium of Claim 34, wherein performing the operation, further comprises getting the bind declaration object of a binding expression.

40. The computer-readable medium of Claim 34, wherein performing the operation, further comprises performing a collection view function.

41. The computer-readable medium of Claim 34, wherein performing the operation, further comprises implementing a collection view that includes checks for context infinity.

42. The computer-readable medium of Claim 34, wherein performing the operation, further comprises supporting object references to objects being used as data context for a binding.

43. The computer-readable medium of Claim 34, wherein performing the operation, further comprises encapsulating arguments for data transfer events, wherein the events are routed events that are handled by a designated handler based on a delegate.

44. The computer-readable medium of Claim 34, wherein performing the operation, further comprises handling a data transfer event raised by a binding.

45. The computer-readable medium of Claim 34, wherein performing the operation, further comprises representing an object reference to an element, with the object reference being specified by its element ID.

46. The computer-readable medium of Claim 45, wherein the object reference is an explicit object reference.

47. The computer-readable medium of Claim 34, wherein performing the operation, further comprises implementing a collection view for collections based on a list.

48. The computer-readable medium of Claim 34, wherein performing the operation, further comprises serving as a data source for data binding.

49. The computer-readable medium of Claim 34, wherein performing the operation, further comprises holding a collection of named parameters.

50. The computer-readable medium of Claim 34, wherein performing the operation, further comprises representing a single select statement to be submitted to a database.

51. The computer-readable medium of Claim 34, wherein performing the operation, further comprises encapsulating arguments passed in an event relating to at least one of an ObjectDataSource, and a RefreshCompleted event of an XmlDataSource.

52. The computer-readable medium of Claim 34, wherein performing the operation, further comprises handling events relating to at least one of a ObjectDataSource.RefreshCompleted event and a XmlDataSource.RefreshCompleted event.

53. The computer-readable medium of Claim 34, wherein performing the operation, further comprises receiving a list of sql commands and names of tables that they should be used to fill.

54. The computer-readable medium of Claim 34, wherein performing the operation, further comprises getting data from a SQL Server for use in databinding.

55. The computer-readable medium of Claim 34, wherein performing the operation, further comprises allowing resource reference to a transformer class that is defined as code-behind in a current application.

56. The computer-readable medium of Claim 34, wherein performing the operation, further comprises declaring namespaces to be used in Xml data binding XPath queries.

57. The computer-readable medium of Claim 34, wherein performing the operation, further comprises serving as a data source for data binding to Extensible Markup Language (XML) content nodes.

58. The computer-readable medium of Claim 34, wherein performing the operation, further comprises declaring an individual namespace within an Extensible Markup Language (XML) data source.

59. The computer-readable medium of Claim 34, wherein performing the operation, further comprises managing a view of a data collection.

60. The computer-readable medium of Claim 34, wherein performing the operation, further comprises handing a CurrentChanged event raised by collection views, or any class implementing the ICurrentItem interface.

61. The computer-readable medium of Claim 34, wherein performing the operation, further comprises representing a method that handles a CurrentChanging event raised by collection view classes, or any class implementing the ICurrentItem interface.

62. The computer-readable medium of Claim 34, wherein performing the operation, further comprises enabling notifications from at least one of the following: items within a collection have changed: an item has been added, removed, or the entire collection has been refreshed.

63. The computer-readable medium of Claim 34, wherein performing the operation, further comprises creating collection view factory classes, which in turn create new CollectionView derived objects.

64. The computer-readable medium of Claim 34, wherein performing the operation, further comprises, maintaining a concept of the current record pointer in a collection view.

65. The computer-readable medium of Claim 34, wherein performing the operation, further comprises creating a One-Way Binding to a Dynamically Updated Data Source.

66. The computer-readable medium of Claim 34, wherein performing the operation, further comprises describing a sort qualifier that is used to sort items in a collection when creating a view.